

Contrôle d'un processeur monocycle

Le document d'accompagnement complète ce sujet de TD.

Vous devez concevoir le contrôle d'un petit processeur. Les instructions sont codées sur 32 bits, les données traitées sont des entiers codés en complément à 2 sur 32 bits. Le processeur possède un banc de 16 registres 32 bits, notés Rx, $0 \leq x \leq 15$, une mémoire de données DMEM et une mémoire d'instructions IMEM, toutes les deux adressées sur 32 bits et mémorisant des mots de 32 bits, et un compteur de programme PC sur 32 bits.

Le chemin de données représenté dans le document d'accompagnement est conçu à partir d'un sous-ensemble du jeu d'instruction comprenant :

- Une instruction de chargement d'une donnée de la mémoire DMEM vers le banc de registre.
- Une instruction de rangement d'une donnée du banc de registre vers la mémoire DMEM.
- Une instruction d'addition et une de soustraction de deux registres avec rangement du résultat dans un 3^e registre.
- Une instruction de multiplication et une de division de deux registres avec rangement du résultat dans un 3^e registre.
- Une instruction de branchement relatif inconditionnel.

La bibliothèque utilisée pour concevoir le processeur inclut les composants suivants :

- Un banc de 16 registres 32 bits.
- Une mémoire adressée sur 32 bits de mots de 32 bits.
- Un extendeur signé en complément à 2 de 16 vers 32 bits.
- Un multiplexeur 2 vers 1 sur 32 bits.
- Une ALU 32 bits pouvant faire une addition ou une soustraction.
- Une unité de Multiplication 32 bits pouvant effectuer une multiplication ou une division.
- Un registre de 32 bascules D.
- Un PLA programmable pouvant réaliser n'importe quelle fonction logique sous forme normale conjonctive avec au maximum 32 entrées et 32 sorties.

Proposez le contrôle du chemin de données représenté dans le document d'accompagnement page 8 : vous donnerez les équations booléennes des signaux de contrôle en fonction des bits de l'instruction.

The Processor Instruction Subset

- add:
 - ADD rd, rs, rt
- sub:
 - SUB rd, rs, rt
- mul:
 - MUL rd, rs, rt
- div:
 - DIV rd, rs, rt
- memory reference:
 - LD rd, [rs, imm16]
 - ST [rs, imm16], rt
- branchement:
 - B imm16

- Registers :
 - rd, rs, rt
- Constantes signées :
 - 16 bits imm16

Instructions : definition et codage

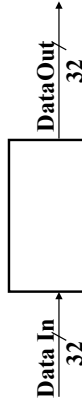
- RTL gives the meaning of the instructions
- All start by fetching the instruction

Inst	Register Transfers															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
#bit	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
ADD	0	1	1	0	rd	rs	rt	imm16								
SUB	0	1	0	0	rd	rs	rt	imm16								
MUL	0	1	0	0	rd	rs	rt	imm16								
DIV	1	0	1	0	rd	rs	rt	imm16								
LD	1	0	0	0	rd	rs	rt	imm16								
ST	0	0	1	1	rd	rs	rt	imm16								
B	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

#bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Instruction	OPCODE				rd				rs				rt				imm16															
ADD	0	1	1	0	rd				rs				rt				0															
SUB	0	1	0	0	rd				rs				rt				0															
MUL	1	0	1	0	rd				rs				rt				0															
DIV	1	0	0	0	rd				rs				rt				0															
LD	0	0	1	1	rd				rs				0				imm16															
ST	0	0	0	1	0				rs				rt				imm16															
B	0	0	1	0	0				0				0				imm16															

PLA

- PLA (idealized)
 - One input bus: Data In
 - One output bus: Data Out
- Function determined by a set of boolean equations in conjunctive normal form



CHEMIN DE DONNEES – P. 8

